Machine Learning Engineer Nanodegree

Capstone proposal

Grasp quality prediction of a robotic hand

Uljan Sinani – 14/07/2020

Domain background

Often, we take for granted our human capability to grasp objects and smoothly manipulate them. As trivial as it may seem for us, it is not the case for a robot. Industrial environments are a common example where we desire the robots to perform tasks close to human capability. Although, robotic manipulators may have superior strength, and might not be endangered from harsh environments, they lack the ability to accurately grasp and handle objects properly. Therefore, to address these issues that robotics field is facing, a machine learning approach could come to the rescue to providing proper domain-related solution.

Problem statement

In industrial environments time and energy are the most important variables to optimize for. The reasons they are important is related to the amount of time it takes for a robot to perform a certain task and as a result a certain amount of energy will get spend performing that action. The latter will get worse if the task must be repeated many times. Increase the number of robots performing poor grasping tasks and we understand how much time translated into energy gets wasted. Based on industrial robots used to manipulate objects, the following are highly desirable.

- Saving energy by performing the action only once
- Grasping effectively the object without causing any damage
- Saving time by quickly grasping the object
- Flexibility, adapting the grasp force according to the nature of the object (i.e. soft, hard, fluffy etc.)

Thus, ensuring a quality grasp of objects from a robot could be a good problem to start from and solve. Tackling this problem, I have selected a dataset coming from a robot company that has simulated a scenario in which a robotic hand tries to grasp an object repeated many times and checking its performance while performing the task. The data collected is acquired using Smart Grasping Sandbox as a virtual environment. The setup contains a universal robotic hand with three joints and an End-effector containing three fingers. A virtual table as a base surface for the object(s) the robot will grasp and a ball being the target object the robot will grasp. Its grasp performance will be based on the surface compression of the object in respect to its original size. The illustration shown below could better help understand the setup conditions under which the data was collected.

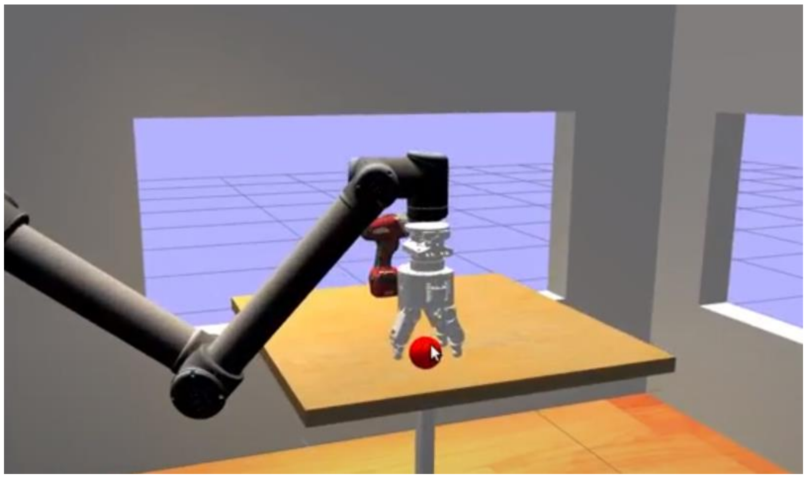
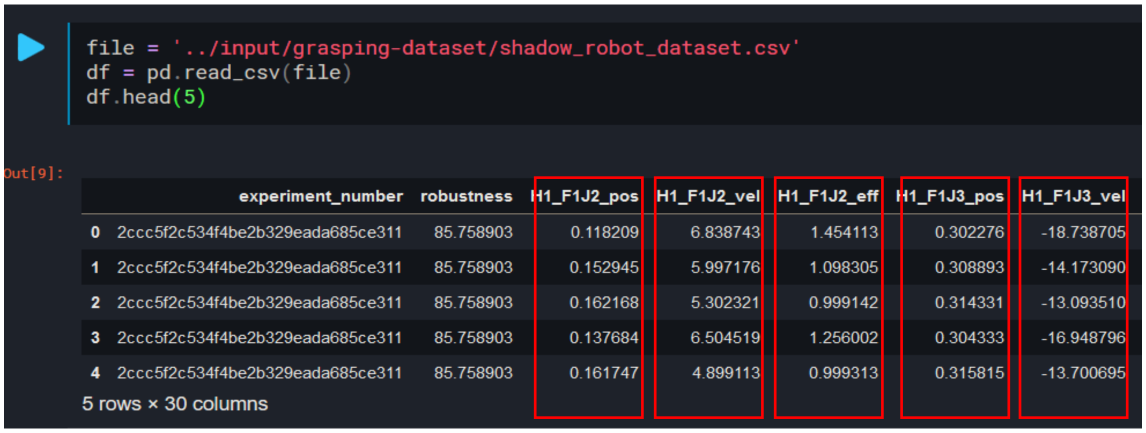


Figure 1 Smart Grasping Sandbox simulation environment

The data is contained in a single dataset called shadow_robot_dataset.csv and it contains 30 columns and in total approximately 1 million rows containing speed, position and robustness values for joints, hands fingers etc. The measurements were obtained while monitoring robot's grasping performance in Smart Grasping Sandbox. An exploration of the data will show that I face values for the position and velocity of robots' fingers that all together contribute to a single robustness value.



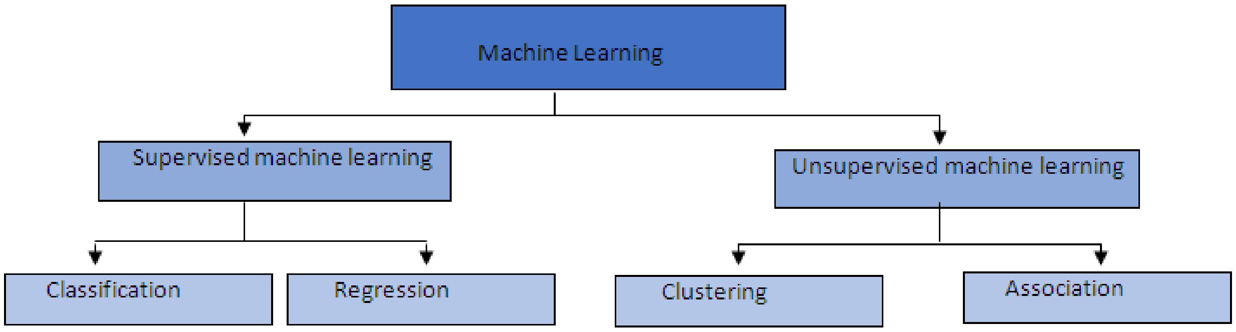
```
file = '../input/grasping-dataset/shadow_robot_dataset.csv'
df = pd.read_csv(file)
df.head(5)
```

Out[9]:

	experiment_number	robustness	H1_F1J2_pos	H1_F1J2_vel	H1_F1J2_eff	H1_F1J3_pos	H1_F1J3_vel
0	2ccc5f2c534f4be2b329eada685ce311	85.758903	0.118209	6.838743	1.454113	0.302276	-18.738705
1	2ccc5f2c534f4be2b329eada685ce311	85.758903	0.152945	5.997176	1.098305	0.308893	-14.173090
2	2ccc5f2c534f4be2b329eada685ce311	85.758903	0.162168	5.302321	0.999142	0.314331	-13.093510
3	2ccc5f2c534f4be2b329eada685ce311	85.758903	0.137684	6.504519	1.256002	0.304333	-16.948796
4	2ccc5f2c534f4be2b329eada685ce311	85.758903	0.161747	4.899113	0.999313	0.315815	-13.700695

5 rows × 30 columns

Hence, dealing with variables that are part of a function that calculates a single output value leads me to a classification problem. The output value that comes out from calculating the variables is based on a function that contributes to a corresponding robustness value, which is the variable of interest. It is calculated as 1/(object _min_distance – construct_coefficient)ˆ2

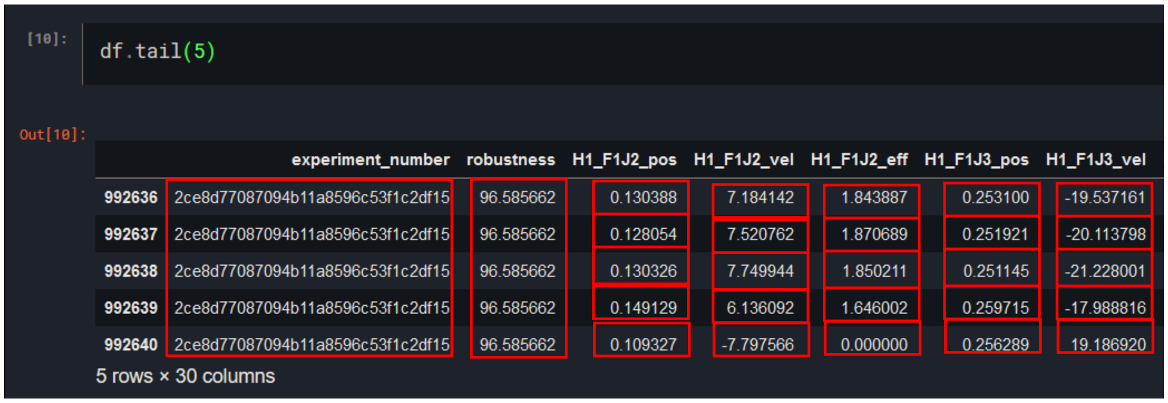


Therefore, given the type of the setup under which the data was gathered, the type of variables and their relationship with the variable(s) of interest means that I am dealing with a classification problem.

Datasets and inputs

The test dataset acquired for this problem is based from Shadow robotics' dataset released in Kaggle. It will serve as input data for my model which comprises of a single CSV file containing test experiments, robustness, and values for hand and finger joints. While exploring the dataset that data that was captured while the robot performed tasks under simulation. Exploring the dataset, I could can see 29 Columns and 992640 rows with numerical values for each hand, and finger joints part of the end-effector. The columns' descriptions appearing in the data set have abbreviated terms related to hand, fingers and joints. For instance, H1 stands for Hand one, F1 for finger one (three fingers per hand), J1 for joint one (three joints per finger), etc.

Looking further, in the dataset obtained from the simulating of the robotic hand in a virtual environment, it is apparent that there are many tasks performed denoted by 'experiment_number' column. It goes to show that for a task the robot performs from start to finish, there are many values recorded of the End-Effector (robotic hand), including fingers and joints in space -based on a reference coordinative system – and their physics of position and velocity. Then, the values for position and velocity in space will condense yield a certain robustness value.



```
[10]:  df.tail(5)

Out[10]:
                  experiment_number  robustness  H1_F1J2_pos  H1_F1J2_vel  H1_F1J2_eff  H1_F1J3_pos  H1_F1J3_vel
992636  2ce8d77087094b11a8596c53f1c2df15   96.585662     0.130388     7.184142     1.843887     0.253100   -19.537161
992637  2ce8d77087094b11a8596c53f1c2df15   96.585662     0.128054     7.520762     1.870689     0.251921   -20.113798
992638  2ce8d77087094b11a8596c53f1c2df15   96.585662     0.130326     7.749944     1.850211     0.251145   -21.228001
992639  2ce8d77087094b11a8596c53f1c2df15   96.585662     0.149129     6.136092     1.646002     0.259715   -17.988816
992640  2ce8d77087094b11a8596c53f1c2df15   96.585662     0.109327    -7.797566     0.000000     0.256289    19.186920
5 rows × 30 columns
```

Figure 2 Dataset content - first five rows

Therefore, it shows that robustness does not change in time after it is calculated once from the initial interaction of the robot with the objects. Practically speaking, robustness in this context, is used to determine the physical character of an object when interacting with another object or interface (i.e. metallic hand of robotic hand with a fluffy ball). Let us imagine a case when we are holding an object, specifically two different balls in our hands. On the left hand we hold a fluffy ball, and on the right hand we are holding a solid ball. Now, imagine throwing at the same time the fluffy ball from the left hand and the solid one from the right hand. Which one would you think will bounce higher? Well, I believe we will agree that fluffy ball from the left hand will bounce higher. In this imaginary experiment, the ball bouncing higher has a low robustness score of its surface, and the one having a hard surface (solid ball) has a higher robustness score. Thus, in the above example the "Robustness" column shows the case when a robot grasps an object and determines its surface character based on the compression force applied when first interacting with it, which will change the theoretical diameter/dimension of the object relative to its center.

Solution statement

To solving the goal achieving a quality grasp it is first important to define what is a good grasp. Kinematically it means the joints of the end effector will have close to zero angular velocity, and zero acceleration compared to neighboring joints relative to the object. The more joints are engaged during the grasp of an object the better the grasp. Having defined the notion of a good grasp and possessed the data the next stage I plan to list the most relevant variables – generated from the simulated sandbox – and then use those for the neural network.

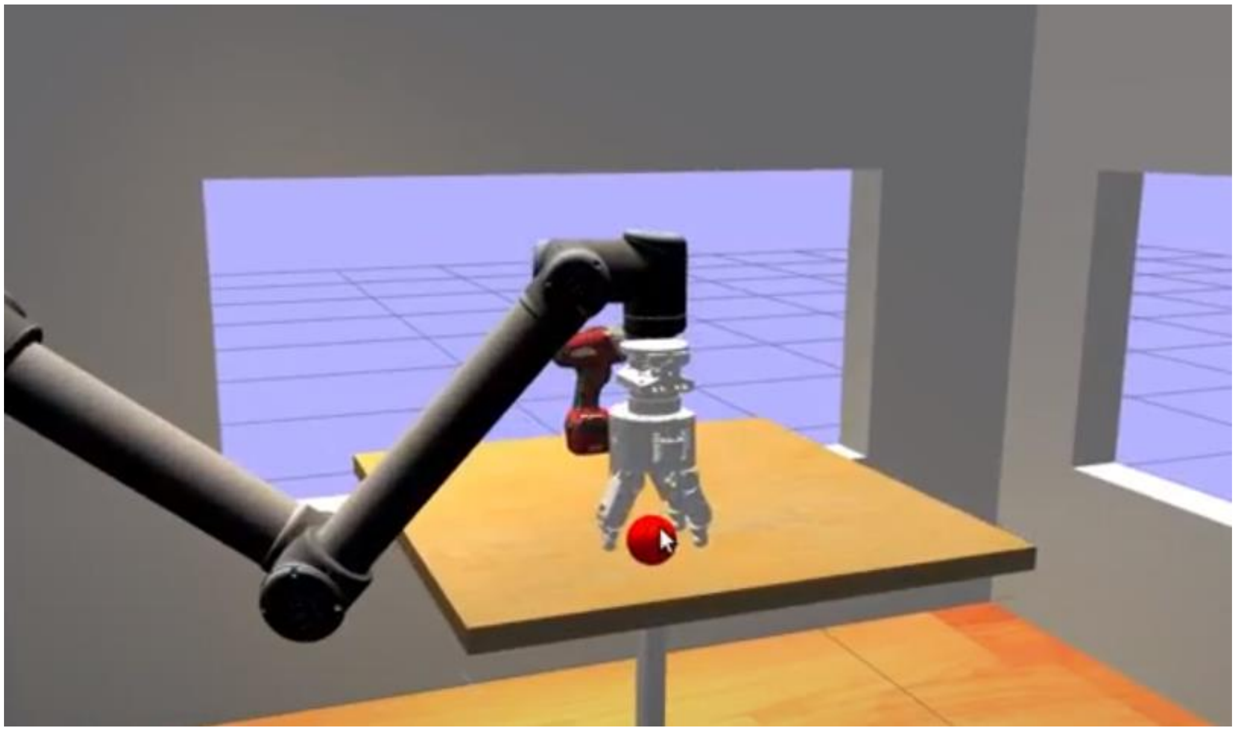


Figure 2 Example of Three-finger End-Effector grasping the object in a Simulated environment

An example of values I plan to take into consideration for the neural network is Angular and linear velocity, and their respective acceleration. These variables do not need to be calculated as they are automatically measured using sensors when the robot motion is simulated in the Smart Grasping Sandbox. However, they will be taken into account as input for our model, to find the relationship between End-Effector joint, and finger position in respect to the robustness value. The model will than get tested based on the accuracy of the value predicted given the data, namely position of joints and fingers in space.

The first network I will try to apply in order estimate the grasp quality will be LSTM neural network. The reason related to the fact that this type of network could accept sampled inputs quantized in time. As such, the dataset provided in Kaggle shows to have the datapoints, time steps and other features which are the variable such as the velocity of finger joints etc.

Benchmark model

For the model benchmark, I plan to initially apply and implement a simple model like LightGBM - which is a gradient boosting framework - and compare it with other Machine Learning libraries such as Keras, SKlearn in order to find which one performs best for a quality grasp. For the final step, I plan to use is LSTM to determine object grasp quality from the End-Effector.

Evaluation Metrics

Accuracy will be the metric which will tell my model how well it performed a certain simulated task when I feed the position, velocity of End-effector joints and fingers. It will allow my model to predict the quality of grasping the object when performing a task. The raw data that will feed the network is expected to be a vector of values from physical position, velocity in space. The output value will be the robustness which will tell how accurate the model was in predicting a good object grasp.

Project design

To start the project, initially I plan to load and explore the nature of the data. Then, the dataset will be checked for any data inconsistencies and/or missing data, in which case they will be corrected. Next, splitting the dataset into train and test files will create four separate sets X_train, Y_train, X_test and Y_test. The following steps will be the creation of the Neural Network Model into which the data will be fed. The final model among others I plan to implement is LSTM, which will take the input data and load them into the workspace to prepare for the training step. The final step I plan is to evaluate the model, where the output will be the predicted grasp quality of the object.

References

[1] Carlos Rubert, Daniel Kappler, Jeannette Bohg and Antonio Morales: Grasp success prediction with quality metrics

[2] Jialiang (Alan) Zhao, Jacky Liang, and Oliver Kroemer: Towards Precise Robotic Grasping by Probabilistic Post-grasp Displacement Estimation

[3] Alex Keith Goins: thesis work

[4] https://www.kaggle.com/ugocupcic/grasp-quality-prediction/

[5] Shadow robot – Building a sandbox for hand-robot training

[6] Google developers – Common ML problems